

**Задорожна Анна Володимирівна**

*кандидат фізико-математичних наук, доцент*

*Львівський національний університет імені Івана Франка*

**Задорожная Анна Владимировна**

*кандидат физико-математических наук, доцент*

*Львовский национальный университет имени Ивана Франко*

**Zadorozhna Anna**

*PhD in Physics and Mathematics, Associate Professor*

*Ivan Franko National University of Lviv*

**Сав'як Назар Тарасович**

*магістрант*

*Львівського національного університету імені Івана Франка*

**Савьяк Назар Тарасович**

*магистрант*

*Львовского национального университета имени Ивана Франко*

**Saviak Nazar**

*Student of the*

*Ivan Franko National University of Lviv*

## ОГЛЯД ФРЕЙМВОРКІВ ТА БІБЛІОТЕК, ЯКІ ВИКОРИСТОВУЮТЬСЯ ПРИ РОЗРОБЦІ ВЕБ-САЙТІВ

## ОБЗОР ФРЕЙМВОРКОВ И БИБЛИОТЕК, ИСПОЛЬЗУЕМЫХ ПРИ РАЗРАБОТКЕ ВЕБ-САЙТОВ

## OVERVIEW OF FRAMEWORKS AND LIBRARIES USED IN WEBSITE DEVELOPMENT

**Анотація.** У статті розглянуто веб-фреймворки та бібліотеки для створення веб-сайтів, сфери їх застосування та переваги використання. Наведено приклади використання фреймворків та бібліотек.

**Ключові слова:** веб-розробка, фреймворк, бібліотека, веб-сайт, фронт-енд, CMS, ReactJS, AngularJS, Skel, MeteorJS, Vue.js, jQuery, BackboneJS, KnockoutJS.

**Аннотация.** В статье рассмотрены веб-фреймворки и библиотеки для создания веб-сайтов, сферы их применения и преимущества использования. Приведены примеры использования фреймворков и библиотек.

**Ключевые слова:** веб-розробка, фреймворк, бібліотека, веб-сайт, фронт-енд, CMS, ReactJS, AngularJS, Skel, MeteorJS, Vue.js, jQuery, BackboneJS, KnockoutJS.

**Summary.** The article is about web frameworks and libraries in creating websites, their scope and the benefits of using them. Examples of using frameworks and libraries are given.

**Key words:** web-development, framework, library, website, front-end, CMS, ReactJS, AngularJS, Skel, MeteorJS, Vue.js, jQuery, BackboneJS, KnockoutJS.

**Постановка проблеми.** Для того щоби створити веб-сайт, необхідно мати як творчі, так і технічні навички. До недавнього часу така робота була досить складною. Використання фреймворку є одним

із основних підходів у створенні веб-сайту. На сьогоднішній день є великий вибір веб-фреймворків та бібліотек, які полегшують та пришвидшують роботу веб-розробника за рахунок уже створених заздалегідь

бібліотек з відкритим вихідним кодом. Фреймворк можна вважати додатковою бібліотекою, але є істотна відмінність. Бібліотеку можна використовувати для розширення функціональності програми, тоді як фреймворк, крім того, визначає архітектуру (взаємозв'язок між компонентами) застосування. Якщо виникає потреба вирішити конкретну проблему, можна використати вже реалізовану для неї бібліотеку. Проте через велику різноманітність постає проблема вибору зручного у використанні фреймворку. Більшість з них написані орієнтовано на певну мову програмування, тому потенціал їх використання та можливості, які вони надають користувачам, потребують подальшого вивчення.

**Аналіз останніх досліджень і публікацій.** Питання про переваги використання фреймворків та бібліотек розглядали на таких сайтах, як web4u, Tproger, WEB Creator, Harb, Web experience та інші.

**Постановка завдання.** Основне завдання статті — розглянути веб-фреймворки та бібліотеки, дати їх детальну характеристику, оцінити переваги та недоліки.

**Виклад основного матеріалу.** Одним із ключових моментів у розвитку всесвітньої павутини є веб-розробка — процес створення веб-сайту. В Інтернеті можна знайти безліч веб-сайтів з відповідями на будь-які запити, проте створення, підтримка та наповнення сайту є важкою працею.

Для того щоб створити новий сайт, необхідно докласти чимало зусиль із різних сторін, а саме інтелектуальних та творчих, адже робота полягає в розробці графічного дизайну та програмного коду на різних мовах веб-програмування, у тому числі мовою HTML, а також, звісно, необхідно створити бази даних та написати текст і розмістити необхідні матеріали, такі, як: фотографії, відео-, аудіоматеріали. Можна з впевненістю сказати, що ці етапи задіюють як інтелектуальні, так і креативні ресурси розробника.

Щоби написати код, потрібні фреймворки та бібліотеки, яких є чимало видів, певні із них будуть розглянуті у роботі. У розробленні веб-сайту основними фреймворками і бібліотеками є:

**jQuery** — достатньо відома бібліотека, яка містить серцевий код. Суть цієї бібліотеки полягає в полегшенні навігації через такий елемент, як DOM, створенні AJAX-застосунків, анімацій та обробки подій. Можливість розміщення плагінів зверху бібліотеки є ще однією перевагою. Саме завдяки цим об'єктам, можна створювати високоефективну анімацію, а також абстракції для взаємодії низького рівня. Ці функції допомагають веб-сторінкам бути більш динамічними та потужними.

**AngularJS** — сьогодні є одним із найпопулярніших фреймворків. Однією із характеристик є відкритість програмного коду. AngularJS застосовується не лише для звичайних веб-додатків, а й динамічних. При цьому використовується HTML для статичних веб-сторінок. Здебільшого він розширює HTML, ре-

алізуючи функції (прив'язка, маніпулювання DOM, перевірка форм на стороні клієнта). Перевагами є: можливість створення та зберігання частин коду, котрі можна використовувати у подальшому, а також утворення нових HTML-елементів та чимала кількість інструментів для текстів.

**ReactJS** — розроблений і підтримується командами Instagram, Facebook достатньо ефективна декларативна бібліотека JavaScript для збірки UI. Основна мета — це допомога в створенні інтерактивно-користувальницького інтерфейсу. У випадку, коли йдеться про розвиток у сфері фронтенд-розробки, саме ця бібліотека буде хорошим початком.

**Vue.js** — доволі реактивний фреймворк, котрий містить кореневу бібліотеку, яка спершу впорається із завданнями уяви, а щодо створення односторінкових додатків, то за це відповідає підсистема додаткових бібліотек.

**BackboneJS** — корисний фреймворк для початківців завдяки своїй легкості в користуванні. Цей фреймворк чудово поєднується з односторінковими додатками, проте для збільшення можливостей доведеться підтягнути інші бібліотеки. Містить багато розширень та плагінів.

**MeteorJS** — підходить для управління базами даних, а також для рендерингу та фронт- і бекенду, адже є доволі функціональним. Навіть більше, для відкриття бази даних підходять пакети лише цієї веб-платформи. Також перевагою є те, що він має достатньо інструментів, щоби створити багатофункціональні додатки.

**Skel** — попри його абсолютну належність до JavaScript, він не є досить популярним. Для створення веб-сайтів та додатків він чудово підійде. Він містить функції API і нормалізації стилів браузера.

**KnockoutJS** — досить легка у використанні JavaScript платформа, яка взаємодіє із моделлю MVVM. Ця платформа має відкритий вихідний код, за її використання немає проблем із прив'язкою даних. Вона добре підходить для односторінкових додатків.

Різниця між фреймворками та бібліотеками полягає в тому, що бібліотека використовується як об'єкт із декількома системами та не впливає на основний програмний продукт. Фреймворк задає розробку архітектури додатку, формуючи на першому етапі поведінку за замовчуванням, формуючи каркас. Фреймворк може включати допоміжні програми, бібліотеки коду, мову сценаріїв та інше програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

Переваги фреймворків:

- розробка допомагає простіше супроводжувати проект на фреймворку;
- бізнес-процеси можуть реалізовуватись із будь-якою послідовністю. Також проекти з основою фреймворк достатньо просто масштабуються та модернізуються;

- щодо рішень та можливості їх навантажень, то тут також є великим плюсом їх швидкість та витривалість. Через це різні інтернет-магазини працюють на фреймворках;
- щодо безпеки, то рішення тут набагато безпечніші, ніж в самописних системах.

Недоліки фреймворків:

- набагато більше часу займає створення функціоналу. Також багато функцій виконуються індивідуально, через те що фреймворки володіють лише певними базовими елементами бізнес-логіки;
- щоб розробляти щось на фреймворку, потрібно мати знання тих бізнес-процесів, над якими працюєте, для прикладу, CMS має певний процес обробки замовлень.

**Висновки.** Отже, ефективність веб-сайту, тобто його практичність, зручність та візуальна привабливість, залежить від виконання певних умов, до яких належать:

- логічне структурування та композиційне впорядкування всіх елементів на веб-сторінках від-

повідно до їх значущості, для досягнення чого визначена необхідність застосування модульної сітки;

- проектування інтерфейсу веб-сайту в такій спосіб, щоби користувач міг легко зрозуміти та розібратись із тим, куди йому необхідно перейти для бажаного результату;
- створення зовнішнього вигляду сайту, який би відповідав змісту, характеру і призначенню сайту;
- дотримання загального стилю або теми на всіх сторінках веб-сайту для того, щоб він сприймався цілісно і гармонійно.

Дотримуватися даних умов допомагають веб-фреймворки й бібліотеки, які надають можливість веб-розробнику не лише зробити сайт сучасним, зручним і привабливим для користувачів, але й розробити даний сайт. Зараз є велика кількість фреймворків і бібліотек, тому кожен розробник зможе знайти для себе зручний і той інструмент, який підходить саме йому.

#### Література

1. Фреймворки в веб разработке. URL: [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks)
2. Сафронов М. Разработка веб-приложений в Yii 2. М.: ДМК Пресс, 2015. 392 с.
3. Сайт Хабрахабр. URL: <https://habr.com/>
4. Сайт Tproger. URL: <https://tproger.ru/>
5. Берд Дж. Веб-дизайн. Руководство разработчика. СПб.: Питер, 2012. 212 с.
6. Знакомство з фреймворками. URL: <http://weblib.com.ua/blog/29>